Mechatronika
BMEGEMIBMMH

Házi Feladat
Hajtáslánc Modellezése

Dobondi-Reisz Mendel
C8GJJW

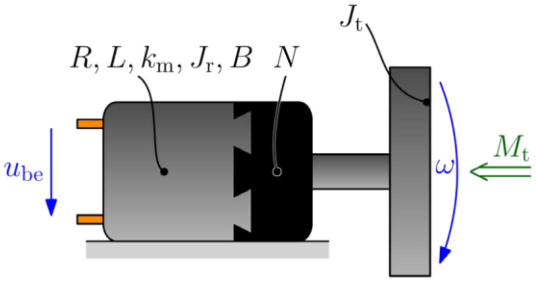
2024.05.11.

Tartalomjegyzék

1. A hajtáslánc modellezése

a) A feladat paraméterei

Az ábrán a MAXON vállalat motorkombinációjának vázlatos felépítése látható. A feladathoz tartozó paraméterlista alapján a vizsgált berendezés az A-MAX 32 típusú, 353242 cikkszámú, 20 W teljesítményű, kefés egyenáramú motorból, valamint a GP 32 BZ, 358335 cikkszámú bolygóműből áll.



1. ábra - Az összeállítás vázlata

A modellezés során feltételezzük, hogy a motor csapágyazásának viszkózus csillapítási tényezője a feladatkiírásban megadott érték: B. Továbbá a modellezés során a tengelykapcsoló tehetetlenségi nyomatékát, valamint a forgó részek torziós rugalmasságát elhanyagoljuk.

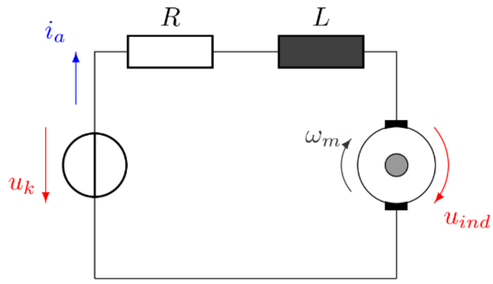
A feladat megoldásához szükséges, előre megadott paramétereket: motor névleges feszültségét, a hajtómű áttételét, a tárcsa tehetetlenségi nyomatékát, valamint a diszkrét idejű számításhoz alkalmazandó időlépést, az alábbi táblázat tartalmazza.

1. táblázat - Adatok

Paraméter neve	Jelölés	Katalógus		SI	
		Érték	Mértékegység	Érték	Mértékegység
Névleges feszültség	u_n	42	$[V]$	42	$[V]$
Névleges áramerősség	i_n	0,659	$[A]$	0,659	$[A]$
Armatúra ellenállás	R	16,6	$[\Omega]$	16,6	$[\Omega]$
Armatúra induktivitás	L	2,22	$[mH]$	$2{,}22 \cdot 10^{-3}$	$[H]$
Motor csillapítása	B	$2 \cdot 10^{-6}$	$\left[\frac{Nms}{rad}\right]$	$2 \cdot 10^{-6}$	$\left[\frac{Nms}{rad}\right]$
Motorállandó	k_m	70,4	$\left[\frac{mN \cdot m}{A}\right]$	0,0704	$\left[\frac{Nm}{A}\right]$
Motor tehetetlenségi nyomatéka	J_r	43,8	$[gcm^2]$	$43{,}8 \cdot 10^{-7}$	$[kgm^2]$
Áttétel	N	35:1	$[-]$	35:1	$[-]$
Tárcsa tehetetlenségi nyomatéka	J_t	200	$[gcm^2]$	$200 \cdot 10^{-7}$	$[kgm^2]$
Diszkrét időlépés	Δt	0,050	$[s]$	0,050	$[s]$

b) A hajtás modellje

A DC motorok állórészében általában állómágnest tartalmaz. Ha a kommutátor kapcsaira feszültséget kapcsolunk, a tekercsben áram fog folyni, amelynek hatására a gerjesztett mágneses tér igyekszik az állórész mágneses mezőjének irányába állni. Forgásba hozva a motor rotorját. Amikor a forgórész elérné a stabil helyzetet, a szénkefék megfordítják az áram irányát, valami a mágneses polaritást, ezért a forgás a „megkezdett” irányba folytatódik.



2. ábra – DC motor modell

Az indukált feszültség és a motor szögsebessége között az alábbi összefüggés teremt kapcsolatot:

$$u_{ind} = k_e \cdot \omega(t) \qquad [1]$$

$$k_e \approx B \cdot l \cdot 2r \cdot cos\big(\varphi(t)\big) \qquad [2]$$

Ahol k_e katalógusban megadott adat, B mágneses indukció, l a mágnesre merőleges áramjárta vezető hossza, r pedig a közöttük lévő távolság fele.

A vezetőben folyó áram és a motor tengelyét forgató nyomaték közötti összefüggés az alábbi alakban írható fel:

$$M_{vill} \equiv M_m = k_m \cdot i(t) \qquad [3]$$

$$k_m \approx 2r \cdot Bl \cdot cos\big(\varphi(t)\big) \qquad [4]$$

Ahol k_e szintén katalógusadat. Ezek az elektromos motor kapcsolóegyenletei. Ezekkel írható le a villamos és mechanikai rendszer közti energiaátalakítás.

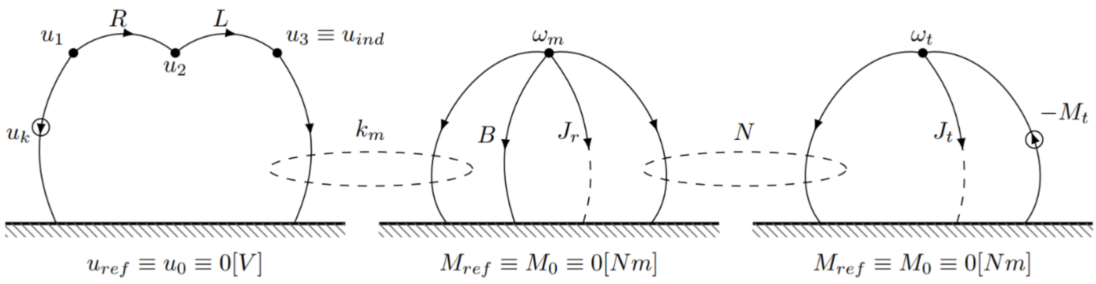
A bolygómű egy áttételt képez a behajtó és kihajtó tengely között. Fontos jellemzője, hogy a forgás irányát nem változtatja meg. Egyben ez azt is jelenti, hogy sem a szögsebesség, sem pedig a nyomaték nem vált előjelet a másik oldalhoz képest. Csupán az áttétel arányainak megfelelően csökkenti vagy növeli az értéket. A bolygómű behajtó tengelye közvetlen kapcsolatban áll motor tengelyével, a kihajtó tengelye pedig közvetlen kapcsolatban áll a tárcsával. Tehát a bolygómű behajtó és kihajtó tengelye közötti kapcsolat, megfelel a motortengely és a tárcsa közötti kapcsolatnak, amely az áttétel ismeretében könnyen felírható. A kapcsolóegyenletek tehát:

$$\omega_m(t) = N \cdot \omega_t(t) \qquad [5]$$

$$M_m(t) = \frac{1}{N} \cdot M_t(t) \qquad [6]$$

Ahol ω_m, a motor szögsebessége, ω_t, a tárcsa szögsebessége, M_m, a motor által leadott nyomaték, M_t, a tárcsa által felvett nyomaték.

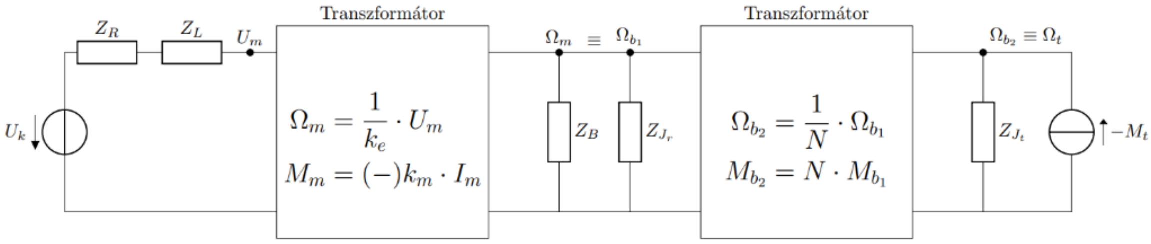
Az energiafolyam rendre a következő elemeken halad keresztül: villamos áramkör, motortengely, tárcsa. A villamos áramkörben modelljében két sorba kötött elem található, egy armatúraellenállás (R) és egy armatúrainduktivitás (L). A motor mechanikai részében két jellemzőt kell figyelembe vennünk: a viszkózus csillapítást (B), valamint a tengely (és egyéb forgó alkatrészek) tehetetlenségi nyomatékát (J_r). A tárcsa esetében is két paramétert kell felírnunk: egyik a tárcsa tehetetlenségi nyomatéka (J_t), a másik pedig az 1. ábrán feltüntetett terhelő nyomaték (M_t). Az egyes szakaszok közötti energiaátalakítókat is figyelembe véve a rendszer struktúragráfja a következő:



3. ábra – A rendszer struktúragráfja

c) A rendszer impedanciahálózatként

A struktúragráf alapján az impedanciahálózat a következőképpen rajzolható fel:



4. ábra – A rendszer impedanciahálózatként

Ahol az egyes impedanciák:

$$Z_R = R \qquad [7]$$

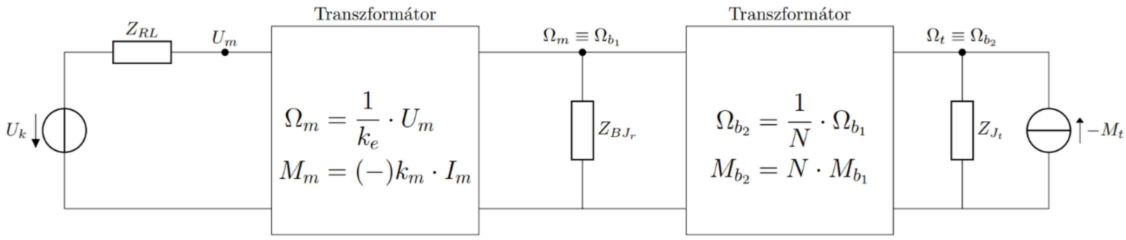
$$Z_L = sL \qquad [8]$$

$$Z_B = \frac{1}{B} \qquad [9]$$

$$Z_{J_r} = \frac{1}{sJ_r} \qquad [10]$$

$$Z_{J_t} = \frac{1}{sJ_t} \qquad [11]$$

Egyes impedanciák az áramköri kapcsolásokra vonatkozó szabályok figyelembevételével összevonhatók, amely során a hálózat képe is megváltozik.



5. ábra – Egyszerűsített impedanciahálózat

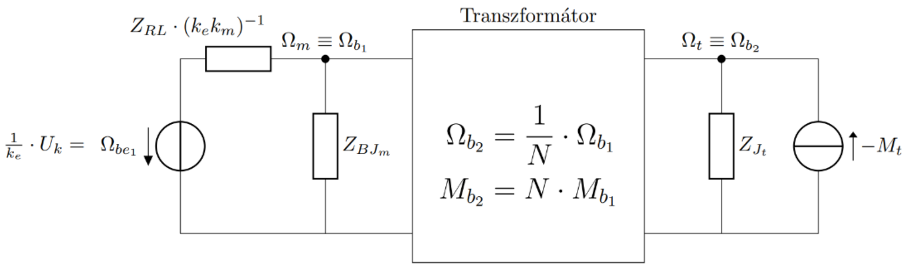
Ahol az egyes összevont impedanciák értékei:

$$Z_{RL} = R + sL \qquad [12]$$

$$Z_{BJ_r} = \frac{1}{B + sJ_r} \qquad [13]$$

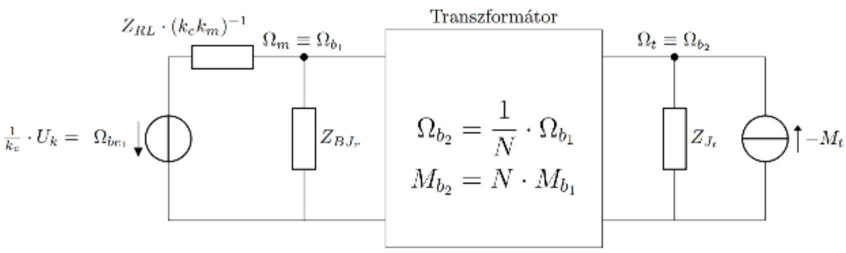
d) Az impedanciahálózat egy oldalra redukálása

Ahhoz, hogy a tárcsa szögsebességét közvetlenül számítani tudjuk a motor kapocsfeszültségéből, arra az oldalra kell redukálnunk az impedanciahálózatot, amely oldalon a keresett érték található. Jelen esetben ez a jobb oldal, tehát először balról minden forrást és impedanciát át kell vinnünk középre. Jelenleg csak transzformátorokkal van dolgunk, ami azt jelenti, hogy ami párhuzamos volt, az párhuzamos marad, ami soros, az pedig soros.



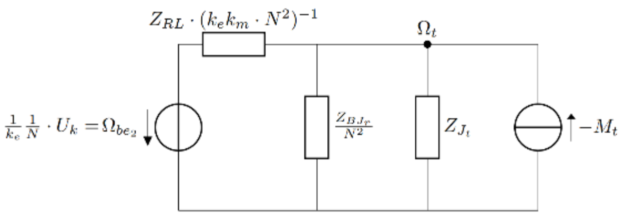
6. ábra – Egyszeresen redukált impedanciahálózat

Ezután megint mindent át kell vinnünk a jobb oldalra a megfelelő kapcsolóegyenletek segítségével.



7. ábra – Egyszerűsített egyszersen redukált impedanciahálózat

Miután a hálózatot a megfelelő oldalra redukáltuk, még egy összevonást megtehetünk.



8. ábra – Egy oldalra redukált impedanciahálózat

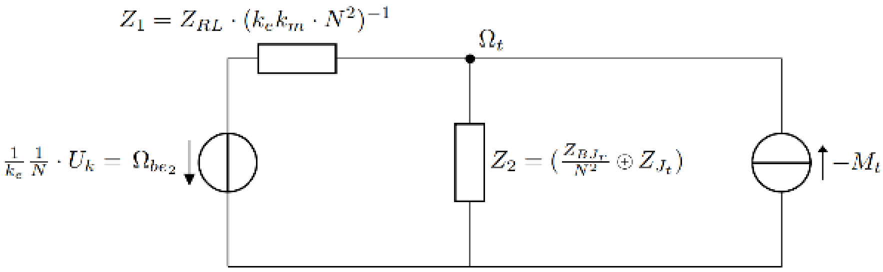
A végső impedanciák értékei:

$$Z_1 = \frac{R + sL}{k_e k_m \cdot N^2} \tag{14}$$

$$Z_2 = \frac{1}{N^2 \cdot B + s(J_t + N^2 J_m)} \tag{15}$$

e) Az átviteli függvény meghatározása

A motor kapocsfeszültsége (u_k) és a tárcsa szögsebessége (ω_t) közti átviteli függvényt keressük. Ebben az esetben a terhelő nyomatékot (M_t) elhanyagoljuk, hiszen számunkra jelenleg csak a kapocsfeszültség hatása érdekes.



9. ábra – Mechanikai oldalra redukált impedanciahálózat

A kimeneti szögsebesség az alábbi módon számítható:

$$\Omega_t(s) = \frac{Z_2}{Z_1 + Z_2} \cdot \Omega_{be_2}(s) = \frac{\frac{1}{N^2 \cdot B + s(J_t + N^2 J_r)}}{\frac{R + sL}{k_e k_m \cdot N^2} + \frac{1}{N^2 \cdot B + s(J_t + N^2 J_r)}} \cdot \frac{1}{k_e} \cdot \frac{1}{N} \cdot U_k(s) \quad [16]$$

$$\Omega_t(s) = \frac{k_m \cdot N}{\left(N^2 \cdot B + s(J_t + N^2 J_r)\right) \cdot (R + sL) + k_e k_m \cdot N^2} \cdot U_k(s) \quad [17]$$

$$W(s) = \frac{\Omega_t(s)}{U_k(s)} = \frac{k_m \cdot N}{\left(N^2 \cdot B + s(J_t + N^2 J_r)\right) \cdot (R + sL) + k_e k_m \cdot N^2} \quad [18]$$

Behelyettesítve a feladat értékeivel, a következő numerikus eredményt kapjuk:

$$W(s) = \frac{2.464}{s^2(3.847260 \cdot 10^{-7}) + s(8.940474 \cdot 10^{-2}) + 6.096047} \quad [19]$$

A függvény pólusai:

$$p_1 = -68.99955 \quad [20]$$

$$p_2 = -7408.933 \quad [21]$$

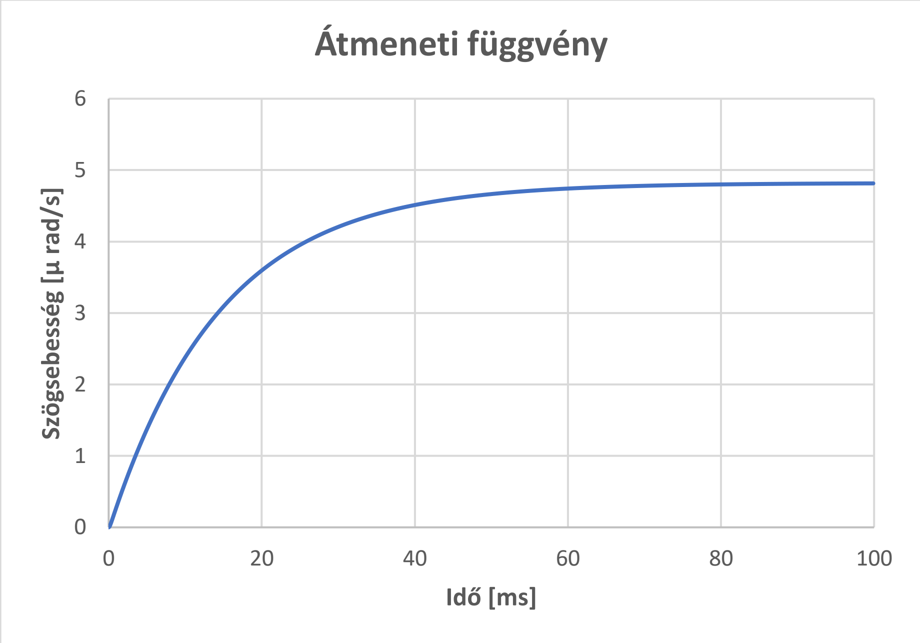
Az átmeneti függvény alakja:

$$w_{u\omega}(t) = \mathcal{L}^{-1}\left\{W(s)\frac{1}{s}\right\} = \frac{k_e N}{p_1(p_1 - p_2)} e^{p_1 t} - \frac{k_e N}{p_2(p_1 - p_2)} e^{p_2 t} + \frac{k_e N}{(p_1 - p_2)} \quad [22]$$

Behelyettesítve:

$$w_{u\omega}(t) = -4.865218 \cdot 10^{-6} \cdot e^{-68.99955t} + 4.530988 \cdot 10^{-8} \cdot e^{-7408.933t} + 4.819908 \cdot 10^{-6} \quad [23]$$

f) Kapocsfeszültség-szögsebesség átmeneti függvény ábrázolása



10. ábra – Átmeneti függvény egységugrás feszültség esetén

g) Modell időállandói

Az időállandók az átviteli függvény pólusaiból könnyen meghatározhatók.

$$T_1 = \frac{1}{|p_1|} = 0.014492\ [s] \tag{24}$$

$$T_2 = \frac{1}{|p_2|} = 1.3497 \cdot 10^{-4}\ [s] \tag{25}$$

2. Analízis

a) További átviteli függvények

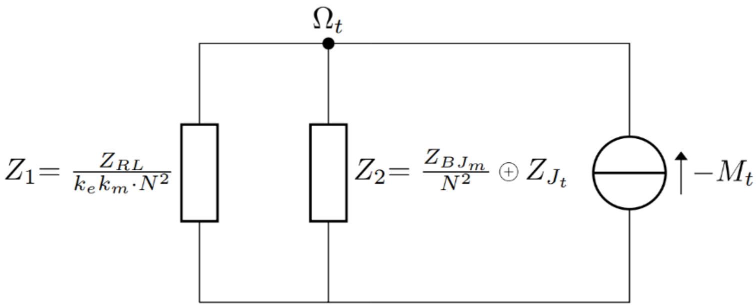
Legyen a bemenet a kapocsfeszültség (U_k) és a tárcsát lassító terhelő nyomaték (M_t), a kimenet pedig a tárcsa szögsebessége (Ω_t) és az armatúrán folyó áram (i).

A kapocsfeszültség és a tárcsa szögsebessége közötti átviteli függvényt már az előző feladatban felírtuk, ami pararaméteresen és numerikusan:

$$W_{U\Omega}(s) = \frac{\Omega_t(s)}{U_k(s)} = \frac{k_m \cdot N}{\left(N^2 \cdot B + s(J_t + N^2 J_r)\right) \cdot (R + sL) + k_e k_m \cdot N^2} \quad [26]$$

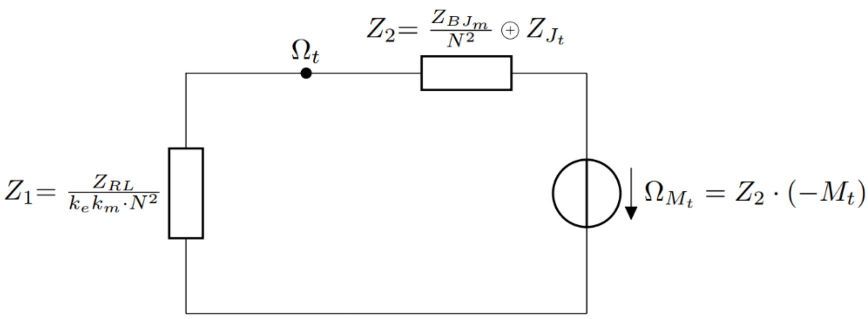
$$W_{U\Omega}(s) = \frac{2.464}{s^2(3.847260 \cdot 10^{-7}) + s(8.940474 \cdot 10^{-2}) + 6.096047} \quad [27]$$

A nyomaték és tárcsa szögsebesség közötti átviteli függvény meghatározásához felhasználhatjuk a jobb oldalra redukált impedanciahálózatot. Mivel most a nyomaték hatása érdekes, ezért az Ω_2 „feszültségforrást” szakadással kell.



11. ábra – A terhelő nyomaték hatása

Ezek után forráscserére lesz szükségünk, mert a keresztváltozót keresünk, de a forrás átmenő változó. Tehát meg kell keresnünk a Norton helyettesítő képet az áramkörben. Ilyen esetben az ideális átmenő változó forráshoz egy párhuzamosan kapcsolt ellenállás tartozik. A belső ellenállás kapcsolási módját felcserélve a Norton képet Thevenin.



12. ábra - Forráscsere

Az új keresztváltozó-forrás értéke az általánosított Ohm-törvény alapján:

$$\Omega_{M_t} = Z_2 \cdot (-M_t) \quad [28]$$

Az nyomaték és szögsebesség közötti kapcsolat:

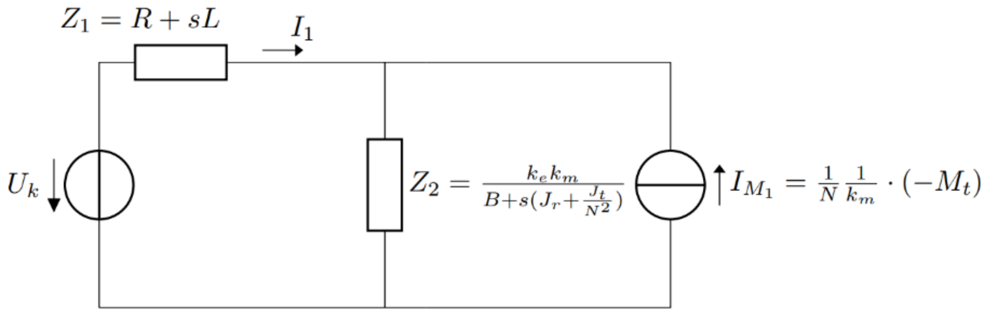
$$\Omega_t = \frac{Z_1}{Z_1 + Z_2} \cdot Z_2 \cdot (-M_t) = (Z_1 \oplus Z_2) \cdot (-M_t) = -(Z_1 \oplus Z_2) \cdot M_t \qquad [29]$$

Az átviteli függvény paraméteresen és numerikusan:

$$W_{M\Omega}(s) = -\frac{sL + R}{(R + sL) \cdot \left(N^2 B + s(N^2 J_r + J_t)\right) + k_e k_m \cdot N^2} \qquad [30]$$

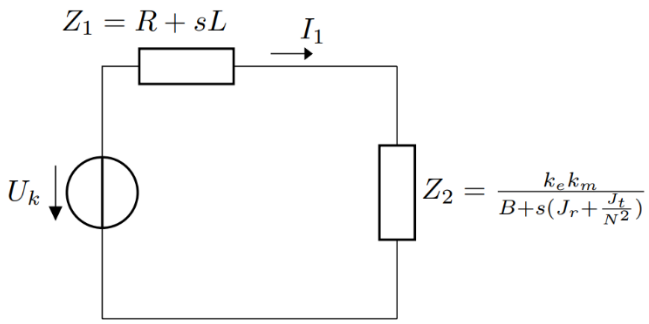
$$W_{M\Omega}(s) = -\frac{s(2.22 \cdot 10^{-3}) + 16.6}{s^2(1.195581 \cdot 10^{-5}) + s(1.437893 \cdot 10^{-2}) + (6.096047)} \qquad [31]$$

Ahhoz, hogy az áramerősségre vonatkozó hatásokat vizsgálni tudjuk, az eredeti impedanciahálózatunkat át kell alakítanunk, mégpedig úgy, hogy az egyes impedanciákat és forrásokat most nem a jobb, hanem a bal oldalra rendezzük.



13. ábra - Villamos oldalra redukált impedanciahálózat

Mivel a feszültségforrás és a terhelő nyomaték hatását külön-külön vizsgáljuk, első körben az áramforrásunkat helyettesítsük egy szakadással. Ekkor az áramkörben csak a feszültségforrás hatása van jelen.



14. ábra - Egyszerűsített impedanciahálózat

Tudjuk, hogy ekkor mindkét impedancián ugyanakkora áram folyik keresztül. Amit az Ohm-törvény felhasználásával egyszerűen ki is tudunk számolni.

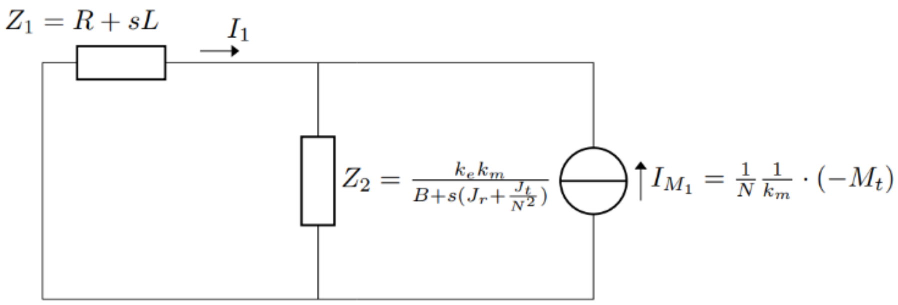
$$I_1 = \frac{1}{Z_1 + Z_2} \cdot U_k = \frac{1}{R + sL + \frac{k_e k_m}{B + s\left(\frac{J_t}{N^2} + J_r\right)}} \cdot U_k \qquad [32]$$

Az átviteli függvény tehát:

$$W_{UI}(s) = \frac{s\left(\frac{J_t}{N^2} + J_r\right) + B}{s^2\left(L\left(\frac{J_t}{N^2} + J_r\right)\right) + s\left(BL + R\left(\frac{J_t}{N^2} + J_r\right)\right) + (BR + k_e k_m)} \qquad [33]$$

$$W_{UI}(s) = \frac{s(4.396327 \cdot 10^{-6}) + 2 \cdot 10^{-6}}{s^2(9.759845 \cdot 10^{-9}) + s(7.298346 \cdot 10^{-8}) + (4.976365 \cdot 10^{-3})} \qquad [34]$$

Ha a nyomaték hatását szeretnénk figyelembe venni, akkor a feszültségforrást kell rövidzárral helyettesíteni.



15. ábra – Terhelő nyomaték hatása

Ebben az esetben az két impedancián különböző áram fog folyni, de azt tudjuk, hogy az értékük fordítottan arányos az impedanciák értékével. Az armatúrán folyó áram tehát:

$$I_1 = \frac{1}{Z_1 + Z_2} \cdot I_{M_1} = \frac{1}{Z_1 + Z_2} \cdot \frac{1}{N}\frac{1}{k_m} \cdot (-M_t) = -\frac{1}{Z_1 + Z_2} \cdot \frac{1}{N}\frac{1}{k_m} \cdot M_t \qquad [35]$$

Az átviteli függvény tehát:

$$W_{MI}(s) = -\frac{k_m N}{\left(J_t L + N^2 J_r L\right)s^2 + \left(J_t R + N^2 J_r R + N^2 BL\right)s + \left(k_e^2 N^2 + N^2 BR\right)} \qquad [36]$$

$$W_{MI}(s) = -\frac{2.006155 \cdot 10^{-3}}{s^2(9.759845 \cdot 10^{-13}) + s(1.173790 \cdot 10^{-8}) + (4.976365 \cdot 10^{-3})} \qquad [37]$$

b) Állandósult állapotok

Az átviteli függvényre vonatkozó végértéktétel alapján, egységnyi bemenetre adott válasz $t \to \infty, s \to 0$ esetén:

$$s \cdot W_{U\Omega}(s) \cdot \frac{1}{s} = \frac{N k_m}{N^2 BR + N^2 k_e k_m} = 0.4041963 \left[\frac{rad}{s}\right] \qquad [38]$$

$$s \cdot W_{M\Omega}(s) \cdot \frac{1}{s} = -\frac{R}{N^2BR + N^2k_ek_m} = -2.723076 \left[\frac{rad}{s}\right] \quad [39]$$

$$s \cdot W_{UI}(s) \cdot \frac{1}{s} = \frac{N^2B}{N^2BR + N^2k_ek_m} = 4.0189998 \cdot 10^{-4}\ [A] \quad [40]$$

$$s \cdot W_{MI}(s) \cdot \frac{1}{s} = \frac{Nk_m}{N^2BR + N^2k_ek_m} = -0.403166\ [A] \quad [41]$$

c) Feszültség-szögsebesség átmeneti függvénye

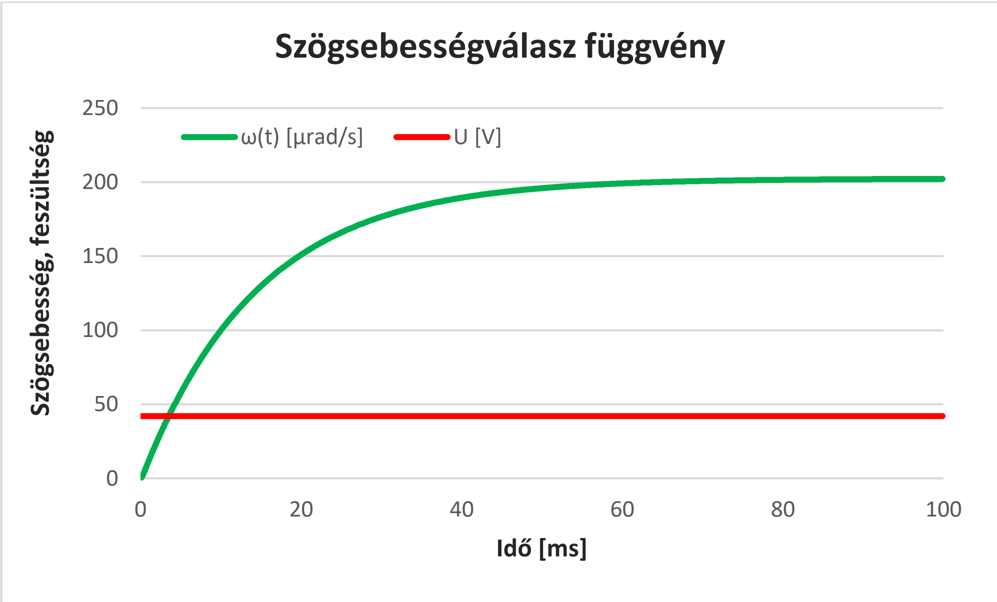
Ha egy rendszerre egységugrás bemenő jelet kapcsolunk, a kimenetén az átmeneti függvény jelenik meg. Legkönnyebben a feszültség-szögsebesség átviteli függvényének segítségével határozhatjuk meg, csupán inverz Laplace-transzformálni kell az átviteli függvény és az egységugrás függvény Laplace transzformáltjának szorzatát.

$$w_{u\omega}(t) = \mathcal{L}^{-1}\left\{W_{U\Omega}(s) \cdot \frac{1}{s}\right\} = \mathcal{L}^{-1}\left\{\frac{k_m \cdot N}{\left(B + s(J_t + N^2 J_r)\right) \cdot (R + sL) + k_e k_m \cdot N^2} \cdot \frac{1}{s}\right\} \quad [42]$$

$$w_{u\omega}(t) = -4.865218 \cdot 10^{-6} \cdot e^{-68.99955t} + 4.530988 \cdot 10^{-8} \cdot e^{-7408.933t} + 4.819908 \cdot 10^{-6} \quad [43]$$

d) Az szögsebesség válasz névleges feszültéség esetén

Ha a rendszer bemenetére névleges feszültséget kapcsolunk, a kimenetén a következő függvény jelenik meg:



16. ábra - Válaszfüggvény névleges feszültség esetén

e) Maximális terhelő nyomaték

A maximális terhelő nyomaték a névleges áram és a nyomaték-áram átviteli függvény állandósult értékének a hányadosaként kapjuk meg.

$$M_{max} = \frac{I_n}{\lim\limits_{s \to 0}\left(s \cdot W_{MI}(s) \cdot \frac{1}{s}\right)} = \frac{0{,}659}{0{,}403166} = 1{,}6347\ [Nm] \quad [44]$$

3. Az egyenáramú motor diszkrét idejű modellezése

a) Állapottér modell előretartó Euler-formulával

Az állapottér modell általánosan az alábbi alakban írható fel, a megadott bemenetek és kimenetnek megfelelően.

$$\begin{bmatrix} x_1[k+1] \\ x_2[k+1] \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1[k] \\ x_2[k] \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \cdot u[k] \qquad [45]$$

$$\begin{bmatrix} y_1[k] \\ y_2[k] \end{bmatrix} = [C_1 \quad C_2] \cdot \begin{bmatrix} x_1[k] \\ x_2[k] \end{bmatrix} + [D] \cdot u[k] \qquad [46]$$

Első körben írjuk fel az áramerősség és feszültség kapcsolatára vonatkozó fizikai egyenleteket.

$$u_k(t) = R \cdot i(t) + L \cdot \frac{di(t)}{dt} + u_{ind}(t) = R \cdot i(t) + L \cdot \frac{di(t)}{dt} + k_e \cdot \omega(t) \qquad [47]$$

Térjünk át diszkrét időbe előre tartó Euler módszerrel.

$$u_k[k] = R \cdot i[k] + L \cdot \frac{i[k+1] - i[k]}{\Delta t} + k_e \cdot \omega[k] \qquad [48]$$

$$\frac{\Delta t}{L} \cdot u_k[k] = \Delta t \frac{R}{L} \cdot i[k] + i[k+1] - i[k] + \Delta t \frac{k_e}{L} \cdot \omega[k] \qquad [49]$$

$$i[k+1] = \left(1 - \Delta t \frac{R}{L}\right) \cdot i[k] - \left(\Delta t \frac{k_e}{L}\right) \cdot \omega[k] + \left(\frac{\Delta t}{L}\right) \cdot u_k[k] \qquad [50]$$

Most írjuk fel az szögsebesség és az áramerősség kapcsolatára vonatkozó fizikai egyenleteket.

$$M(t) = k_m \cdot i(t) = J_r \frac{d\omega(t)}{dt} + B\omega(t) \qquad [51]$$

Az előzően hasonlóan áttérünk diszkrét időbe.

$$k_m \cdot i[k] = J_r \frac{\omega[k+1] - \omega[k]}{\Delta t} + B\omega[k] \qquad [52]$$

$$k_m \cdot i[k] = \frac{J_r}{\Delta t} \omega[k+1] - \frac{J_r}{\Delta t} \omega[k] + B\omega[k] \qquad [53]$$

$$\Delta t \frac{k_m}{J_r} \cdot i[k] = \omega[k+1] - \omega[k] + \Delta t \frac{B}{J_r} \omega[k] \qquad [54]$$

$$\omega[k+1] = \left(\Delta t \frac{k_m}{J_r}\right) \cdot i[k] + \left(1 - \Delta t \frac{B}{J_r}\right) \omega[k] \qquad [55]$$

Ezzel az állapotegyenleteink megvannak, már csak mátrix egyenletté kell őket alakítani.

$$\begin{bmatrix} i[k+1] \\ \omega[k+1] \end{bmatrix} = \begin{bmatrix} \left(1-\Delta t\frac{R}{L}\right) & \left(-\Delta t\frac{k_e}{L}\right) \\ \left(\Delta t\frac{k_m}{J_r}\right) & \left(1-\Delta t\frac{B}{J_r}\right) \end{bmatrix} \cdot \begin{bmatrix} i[k] \\ \omega[k] \end{bmatrix} + \begin{bmatrix} \left(\frac{\Delta t}{L}\right) \\ 0 \end{bmatrix} \cdot u_k[k] \qquad [56]$$

$$y[k] = [0 \quad 1] \cdot \begin{bmatrix} i[k] \\ \omega[k] \end{bmatrix} + [0] \cdot u_k[k] \qquad [57]$$

A modell numerikusan:

$$\begin{bmatrix} i[k+1] \\ \omega[k+1] \end{bmatrix} = \begin{bmatrix} -372{,}8739 & -1.585586 \\ 803.6530 & 0{,}9771689 \end{bmatrix} \cdot \begin{bmatrix} i[k] \\ \omega[k] \end{bmatrix} + \begin{bmatrix} 22{,}52252 \\ 0 \end{bmatrix} \cdot u_k \qquad [58]$$

$$y[k] = [0 \quad 1] \cdot \begin{bmatrix} i[k] \\ \omega[k] \end{bmatrix} + [0] \cdot u_k \qquad [59]$$

b) Állapottér modell hátra tartó Euler módszerrel

A hátra tartó Euler modell paramétereit számolhatjuk az előre tartó modell mátrixaiból.

Forward Euler		Backward Euler	
$\boldsymbol{A_{fe}} = \boldsymbol{I} + \boldsymbol{A} \cdot \Delta t$	[60]	$\boldsymbol{A_{be}} = \left(\boldsymbol{2I} - \boldsymbol{A_{fe}}\right)^{-1}$	[64]
$\boldsymbol{B_{fe}} = \boldsymbol{B} \cdot \Delta t$	[61]	$\boldsymbol{B_{be}} = \boldsymbol{A_{be}} \cdot \boldsymbol{B_{fe}}$	[65]
$\boldsymbol{C_{fe}} = \boldsymbol{C}$	[62]	$\boldsymbol{C_{be}} = \boldsymbol{C_{fe}} \cdot \boldsymbol{A_{fe}}$	[66]
$\boldsymbol{D_{fe}} = \boldsymbol{D}$	[63]	$\boldsymbol{D_{be}} = \boldsymbol{C_{be}} \cdot \boldsymbol{B_{fe}} + \boldsymbol{D_{fe}}$	[67]

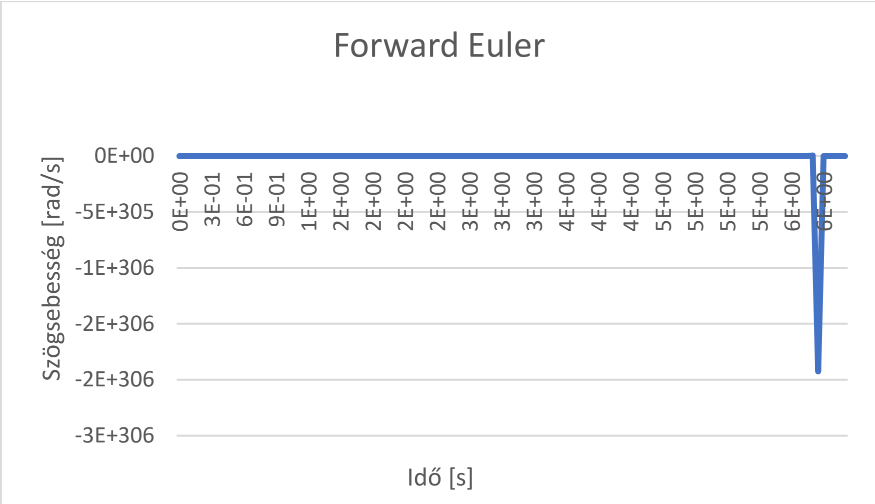
Behelyettesítve a programból kapott adatokat:

$$\begin{bmatrix} i[k+1] \\ \omega[k+1] \end{bmatrix} = \begin{bmatrix} 0.0006170207 & -0.0009565012 \\ 0.4848020 & 0.2261419 \end{bmatrix} \cdot \begin{bmatrix} i[k] \\ \omega[k] \end{bmatrix} + \begin{bmatrix} 0.01389686 \\ 10.91896 \end{bmatrix} \cdot u_k \qquad [68]$$

$$y[k] = [0.4848020 \quad 0.2261419] \cdot \begin{bmatrix} i[k] \\ \omega[k] \end{bmatrix} + [10.91896] \cdot u_k \qquad [69]$$

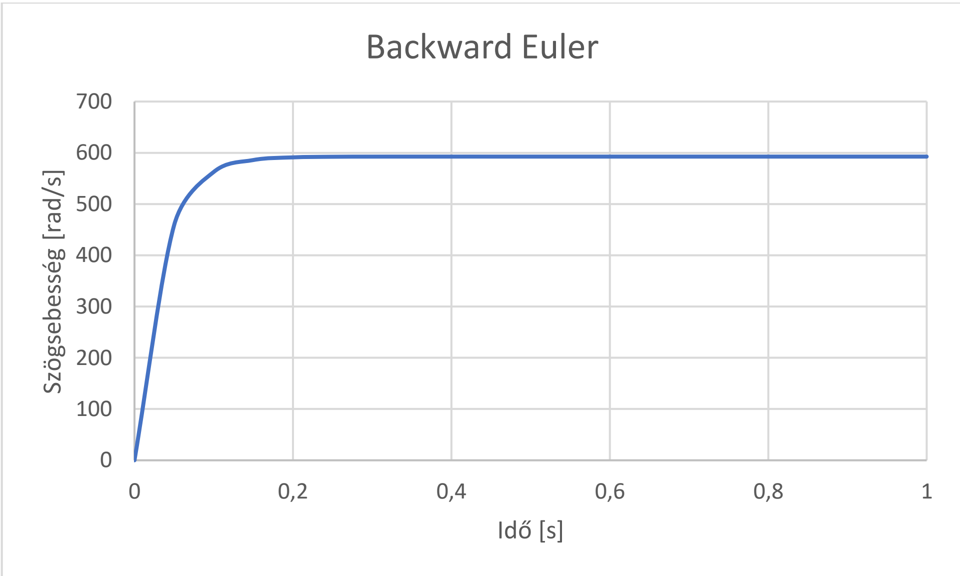
c) Egységugrás gerjesztés hatása

Az előretartó Euler módszerrel kapott eredmény, oszcillálva a pozitív és negatív tartomány között, a végtelenbe divergál.



17. ábra - Diszrét kimeneti függvény (Forward Euler)

Ahogy az ábrán is láthatjuk, a szögsebesség az hátra tartó Euler módszer esetén a megfelelő értékre konvergál.



18. ábra – Diszkrét kimeneti függvény (Backward Euler)

4. Paraméterváltozás hatása

A paraméterváltozás hatását a maximális terhelő nyomatékra könnyen meghatározhatjuk a korábbi számításaink felhasználásával. A B paraméter értékét kell csak átírni a programkódban. Az eredmények:

2. táblázat – Motor viszkózus csillapításának hatása

Megnevezés	Mértékegység	$B \cdot 0{,}1$	B	$B \cdot 10$
B_{uj}	$[Nms/rad]$	$2 \cdot 10^{-7}$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$
$M_{t_{max}}$	$[Nm]$	1,314935	1,592892	1,620688

5. Feladat megoldásához használt C programkód

```
#include <stdio.h>
#include <math.h>
#include <string.h>



void save_to_csv(const char *filename, double *t, char* t_name, double *y,
char* y_name, int size) {
    FILE *file = fopen(filename, "w");
    if (!file) {
        perror("Failed to open file");
        return;
    }
    fprintf(file, t_name);
    fprintf(file, ",");
    fprintf(file, y_name);
    fprintf(file, "\n");
    for (int i = 0; i < size; ++i)
    {
        fprintf(file, "%.6e,%.6e\n", t[i], y[i]);
    }
    fclose(file);
}
void matrixVectorMultiply(double matrix[2][2], double vector[2], double
result[2]) {
    result[0] = matrix[0][0] * vector[0] + matrix[0][1] * vector[1];
    result[1] = matrix[1][0] * vector[0] + matrix[1][1] * vector[1];
}
void vectorMatrixMultiply(double vector[2], double matrix[2][2], double
result[2]) {
    result[0] = vector[0] * matrix[0][0] + vector[1] * matrix[1][0];
    result[1] = vector[0] * matrix[0][1] + vector[1] * matrix[1][1];
}
void vectorVectorMultiply(double row_vector[2], double column_vector[2],
double *result) {
    *result = row_vector[0] * column_vector[0] + row_vector[1] *
column_vector[1];
}
void invertMatrix(double A[2][2], double A_inv[2][2]) {
    double determinant = A[0][0] * A[1][1] - A[0][1] * A[1][0];

    if (determinant == 0) {
        printf("Error: Matrix is singular, cannot be inverted.\n");
        return;
```

```
        }

        double inv_det = 1.0 / determinant;

        A_inv[0][0] = A[1][1] * inv_det;
        A_inv[0][1] = -A[0][1] * inv_det;
        A_inv[1][0] = -A[1][0] * inv_det;
        A_inv[1][1] = A[0][0] * inv_det;
    }
    void changeDelimiter(const char *original_delim, const char *new_delim,
const char *filename) {
        FILE *file = fopen(filename, "r");
        if (file == NULL)
        {
            printf("Error: Unable to open file %s\n", filename);
            return;
        }

        FILE *tempFile = tmpfile();
        if (tempFile == NULL)
        {
            printf("Error: Unable to create temporary file\n");
            fclose(file);
            return;
        }

        char line[1000];
        while (fgets(line, sizeof(line), file) != NULL)
        {
            char *delim = strstr(line, original_delim);
            while (delim != NULL)
            {
                memcpy(delim, new_delim, strlen(new_delim));
                delim = strstr(delim + strlen(new_delim), original_delim);
            }
            fputs(line, tempFile);
        }

        fclose(file);
        rewind(tempFile);

        file = fopen(filename, "w");
        if (file == NULL)
        {
            printf("Error: Unable to open file %s\n", filename);
            fclose(tempFile);
            return;
        }
```

```
    int c;
    while ((c = fgetc(tempFile)) != EOF)
    {
        fputc(c, file);
    }

    fclose(tempFile);
    fclose(file);
}




int main()
{
    printf("\n");
    printf(" _____________________________________________ \n");
    printf("|                                             |\n");
    printf("|          MECHATRONIKA HAZI FELADAT          |\n");
    printf("|             DC MOTOR MODELLEZES             |\n");
    printf("|_____________________________________________|\n");
    printf("\n\n\n\n\n");





    // ##################### ADATOK ###################

    // Aramkor parameterek
    double u_n = 42;        // [V]
    double i_n = 0.659;     // [A]
    double R = 16.6;        // [ohm]
    double L = 2.22e-3;     // [H]
    // Motor parameterek
    double B = 2e-6;        // [Nms/rad]
    double k_m = 70.4e-3;   // [Nm/A]
    double J_r = 43.8e-7;   // [kgm2]
    // Bojgomu parameterek
    double N = 35;             // [-]
    double J_t = 200e-7;    // [kgm2]
    // Egyeb
    double dt = 0.050;      // [s]

    // Adatok kiiratasa
    printf("========== ADATOK ==========\n");
    printf("\n\tAramkor\n");
    printf("\t\tR  = %E [ohm]\n", R);
    printf("\t\tL  = %E [H]\n", L);
```

```
        printf("\tMotor\n");
        printf("\t\tB  = %E [Nms/rad]\n", B);
        printf("\t\tJr = %E [kgm2]\n", J_r);
        printf("\t\tkm = %E [Nm/A]\n", k_m);
        printf("\tBolygomu\n");
        printf("\t\tN  = %E [-]\n", N);
        printf("\tTarcsa\n");
        printf("\t\tJt = %E [kgm2]\n", J_t);
        printf("\n\n\n\n\n");





        // ############# UO ATVITELI FUGGVENY #############

        /*
            ss * b0 + s * b1 + 1 * b2
            -------------------------
            ss * a0 + s * a1 + 1 * a2
        */

        // Atviteli fuggveny parameterek
        double b0 = k_m * N;
        double a2 = J_t * L + N * N * J_r * L;
        double a1 = J_t * R + N * N * J_r * R + N * N * B * L;
        double a0 = k_m * k_m * N * N + N * N * B * R;
        printf("========== UO ATVITELI FUGGVENY ==========\n");
        printf("\n\tW_UO(s) = ____ss(%E)_+_s(%E)_+_(%E)____\n", 0.0, 0.0, b0);
        printf(  "\t              ss(%E) + s(%E) + (%E)\n", a2, a1, a0);

        // Diszkriminans
        double D = a1 * a1 - 4 * a0 * a2;

        // Polusok
        double p1 = (-a1 + sqrt(D)) / (2 * a2);
        double p2 = (-a1 - sqrt(D)) / (2 * a2);
        printf("\n\tA rendszer polusai:\n");
        printf("\t\tp1 = %E\n", p1);
        printf("\t\tp2 = %E\n", p2);

        // Atviteli fuggveny egyutthatoi
        double v1 = b0 / ((p1 - p2) * p1);
        double v2 = -b0 / ((p1 - p2) * p2);
        double v3 = b0 / (p1 * p2);
        printf("\n\tAz atviteli fuggveny alakja:\n");
        printf("\n\t\tw_uo(t) = %E * exp(%E * t) + %E * exp(%E * t) + %E\n",
v1, p1, v2, p2, v3);
```

```
        // Idoallandok
        double T1 = 1 / fabs(p1);
        double T2 = 1 / fabs(p2);
        printf("\n\tAz atviteli fuggveny idoallandoi:\n");
        printf("\t\tT1 = %E [s]\n\t\tT2 = %E [s]\n", T1, T2);
        printf("\n\n\n\n\n");





        // ########### ATVITELI FUGGVENYEK ALLANDOSULT ALLAPOTAI ###########

        double W_UO_Stat = k_m * N / a0;
        double W_MO_Stat = N * N * B / a0;
        double W_UI_Stat = -R / a0;
        double W_MI_Stat = k_m * N / a0;
        printf("========== ATVITELI FUGGVENYEK ALLANDOSULT ALLAPOTAI
==========\n");
        printf("\n\tlim(s->0)  W_UO(s) = %E [rad/s]\n\tlim(s->0)  W_UI(s) = %E
[rad/s]\n\tlim(s->0)  W_MO(s) = %E [A]\n\tlim(s->0)  W_MI(s) =
%E\n",  W_UO_Stat, W_MO_Stat, W_UI_Stat, W_MI_Stat);
        printf("\n\n\n\n\n");





        // ########### MAXIMALIS TERHELO NYOMATEK ###########

        double M_t_max = (i_n - W_MO_Stat * u_n) / W_MI_Stat;
        printf("========== MAXIMALIS TERHELO NYOMATEK ==========\n");
        printf("\n\tM_t_max = %E [Nm]\n", M_t_max);
        printf("\n\n\n\n\n");





        // ########### ALLAPOTTER MODELL ###########

        // Elore tarto Euler
        double A_fe[2][2] = {
            {1 - (dt * R / L), -dt * k_m / L},
            {k_m * dt / J_r, 1 - (B * dt) / J_r}
        };
        double B_fe[2] = {dt / L, 0};
        double C_fe[2] = {0, 1};
        double D_fe = 0;
```

```
printf("========== ALLAPOTTER MODELL ==========\n");
printf("\n\tElore tarto (forward) Euler-formulaval\n");
printf("\t\t     _                                   _ \n");
printf("\t\tA = |   %14.6E   %14.6E   |\n", A_fe[0][0], A_fe[0][1]);
printf("\t\t    |_  %14.6E   %14.6E  _|\n", A_fe[1][0], A_fe[1][1]);
printf("\t\t     _                 _ \n");
printf("\t\tB = |   %14.6E   |\n", B_fe[0]);
printf("\t\t    |_  %14.6E  _|\n", B_fe[1]);
printf("\n");
printf("\t\tC = [   %14.6E   %14.6E   ]\n", C_fe[0], C_fe[1]);
printf("\n");
printf("\t\tD = [   %14.6E   ]\n", D_fe);

// Hatra tarto Euler (be ~ backward Euler)
double E_be[2][2] = {{1,0}, {0,1}};
double A_be_inv[2][2];
double A_be[2][2];
double B_be[2];
double C_be[2];
double D_be;
for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 2; j++)
    {
        A_be_inv[i][j] = 2 * E_be[i][j] - A_fe[i][j];
    }
}
invertMatrix(A_be_inv, A_be);
matrixVectorMultiply(A_be, B_fe, B_be);
vectorMatrixMultiply(C_fe, A_be, C_be);
vectorVectorMultiply(C_fe, B_be, &D_be);
printf("\n\tHatra tarto (backward) Euler-formulaval\n");
printf("\t\t     _                                   _ \n");
printf("\t\tA = |   %14.6E   %14.6E   |\n", A_be[0][0], A_be[0][1]);
printf("\t\t    |_  %14.6E   %14.6E  _|\n", A_be[1][0], A_be[1][1]);
printf("\t\t     _                 _ \n");
printf("\t\tB = |   %14.6E   |\n", B_be[0]);
printf("\t\t    |_  %14.6E  _|\n", B_be[1]);
printf("\n");
printf("\t\tC = [   %14.6E   %14.6E   ]\n", C_be[0], C_be[1]);
printf("\n");
printf("\t\tD = [   %14.6E   ]\n", D_be);
printf("\n\n\n\n\n");




// ########### GRAFIKON ADATOK KIMENTESE ###########
```

```
        // w_uo atmeneti fuggveny
        int data_points = 1e3;
        double t[data_points];
        double w_uo[data_points];
        for (int i = 0; i < data_points; ++i) {
            t[i] = i * 1e-4;
            w_uo[i] = v1 * exp(p1 * t[i]) + v2 * exp(p2 * t[i]) + v3;
        }
        printf("========== ADATOK MENTESE ==========\n");
        save_to_csv("w_uo_atmeneti_fgv.csv", t, "t",w_uo, "w_uo",
data_points);
        printf("\n\tA w_uo atmeneti fuggveny el van
mentve!\n\t\tw_uo_atmeneti_fgv.csv\n");

        // Eloretarto (forward) Euler
        double y[data_points];
        double u[data_points];
        for (int i = 0; i < data_points; ++i) {
            u[i] = 42;
        }
        double X_1[2] = {0, 0};
        double X_0[2] = {0, 0};
        for (int k = 0; k < data_points; ++k) {
            X_1[0] = A_fe[0][0] * X_0[0] + A_fe[0][1] * X_0[1] + B_fe[0] *
u[k];
            X_1[1] = A_fe[1][0] * X_0[0] + A_fe[1][1] * X_0[1] + B_fe[1] *
u[k];
            y[k] = C_fe[0] * X_0[0] + C_fe[1] * X_0[1] + D_fe * u[k];
            t[k] = k*dt;
            X_0[0] = X_1[0];
            X_0[1] = X_1[1];
        }
        save_to_csv("forward_euler.csv", t, "t [s]", y, "O [rad/s]",
data_points);
        printf("\n\tA w_uo atmeneti fuggveny el van
mentve!\n\t\tforward_euler.csv\n");

        // Hatratarto (backward) Euler
        for (int i = 0; i < 2; i++)
        {
            X_0[i] = 0;
            X_1[i] = 0;
        }

        for (int k = 0; k < data_points; ++k)
        {
            X_1[0] = A_be[0][0] * X_0[0] + A_be[0][1] * X_0[1] + B_be[0] *
u[k];
```

```
            X_1[1] = A_be[1][0] * X_0[0] + A_be[1][1] * X_0[1] + B_be[1] *
u[k];
            y[k] = C_be[0] * X_0[0] + C_be[1] * X_0[1] + D_be * u[k];
            t[k] = k*dt;
            X_0[0] = X_1[0];
            X_0[1] = X_1[1];
        }
        save_to_csv("backward_euler.csv", t, "t [s]", y, "O [rad/s]",
data_points);
        printf("\n\tA w_uo atmeneti fuggveny el van
mentve!\n\t\tbackward_euler.csv\n");
        printf("\n\n\n\n\n");

        // CSV MAGYARRA ALLITASA, MERT NEM KEPES RA AZ EXCEL
        changeDelimiter(",",";", "w_uo_atmeneti_fgv.csv");
        changeDelimiter(".",",", "w_uo_atmeneti_fgv.csv");
        changeDelimiter(",",";", "forward_euler.csv");
        changeDelimiter(".",",", "forward_euler.csv");
        changeDelimiter(",",";", "backward_euler.csv");
        changeDelimiter(".",",", "backward_euler.csv");




        // ########### PARAMETERVALTOZAS HATASA ###########
        double tmp = B;
        B = B*10;
        a0 = k_m * k_m * N * N + N * N * B * R;
        W_UO_Stat = k_m * N / a0;
        W_MO_Stat = N * N * B / a0;
        W_UI_Stat = -R / a0;
        W_MI_Stat = k_m * N / a0;
        printf("========== B PARAMETER VALTOZASANAK HATASA ==========\n");
        printf("\nB * 0.1 = 2E-7 [Nms/rad] eseten:\n");
        M_t_max = (i_n - W_MO_Stat * u_n) / W_MI_Stat;
        printf("\n\tM_t_max = %E [Nm]\n", M_t_max);
        B = tmp;
        a0 = k_m * k_m * N * N + N * N * B * R;
        W_UO_Stat = k_m * N / a0;
        W_MO_Stat = N * N * B / a0;
        W_UI_Stat = -R / a0;
        W_MI_Stat = k_m * N / a0;
        printf("\n");

        printf("\nB * 1 = 2E-6 [Nms/rad] eseten:\n");
        M_t_max = (i_n - W_MO_Stat * u_n) / W_MI_Stat;
        printf("\n\tM_t_max = %E [Nm]\n", M_t_max);
        B = tmp/10;
```

```
        a0 = k_m * k_m * N * N + N * N * B * R;
        W_UO_Stat = k_m * N / a0;
        W_MO_Stat = N * N * B / a0;
        W_UI_Stat = -R / a0;
        W_MI_Stat = k_m * N / a0;
        printf("\n");

        printf("\nB * 10 = 2E-5 [Nms/rad] eseten:\n");
        M_t_max = (i_n - W_MO_Stat * u_n) / W_MI_Stat;
        printf("\n\tM_t_max = %E [Nm]\n", M_t_max);
        printf("\n\n\n\n\n");





        // ########### VEGE ###########
        printf("\n");
        printf(" ______________________________________________ \n");
        printf("|                                              |\n");
        printf("|             DOBONDI-REISZ MENDEL             |\n");
        printf("|                    C8GJJW                    |\n");
        printf("|                 2024.05.16.                  |\n");
        printf("|______________________________________________|\n");
        printf("\n\n\n\n\n");

        return 0;
    }
```